Amendments to the Specification

Replace paragraphs **[0024]** and **[0025]** with the following paragraphs:

[0024] Scannable microcode register array **20** is comprised of a plurality of serially connected instruction shift-registers. For purposes of illustration, ~~seven~~ eight registers are shown in **Figure 3**. The number of instructions in the array depends on the specific design requirements and design budget. Each register may be in the order of 34 or more bits and stores a microcode test instruction for execution. The instructions are "ordered" which means that the instructions are executed in sequence. The first address is instruction zero. The instructions stored in the registers perform tests of the memory according to predetermined test algorithms under the control of the pointer controller and the repeat loop module and in accordance with command fields contained in the instructions. An objective of the invention is to enable one to scan in instructions to perform memory tests according to virtually any test algorithm. Thus, the details of the algorithm and of the manner in which data pattern and address sequencing is achieved is not important for the purposes of the present invention and, accordingly, are not described herein.

[0025] Test instructions are serially loaded into the scannable microcode register array ~~16~~ 20 via test interface **12**. One instruction is executed for each execution of an operation applied to the memory under test by the sequencer. The microcode instructions of the memory BIST controller provide parallel control of blocks such as the address generator, data generator, sequencer and pointer controller, creating a wide but very flexible architecture for the generation of complex test algorithms. The contents of the test instructions in the scannable microcode register array are not modified during a test. The pointer controller, described in more detail later, operates to select an instruction for execution and determines the branch for execution of the next instruction from data contained in each test instruction. Before describing the structure and operation of the various sub-circuits of the test controller, it would be useful to briefly describe the various instruction fields which comprise a microcode instruction. It is to be understood at the outset that additional fields may be provided without departing from the present invention.

Replace paragraphs **[0042]** with the following paragraph:

[0042] It will be understood that other ~~the~~ field decodes may be altered from that shown above and that other fields may be included in the instructions without departing from the spirit of the present invention.

Replace paragraphs **[0046]** with the following paragraph:

[0046] Branch To Instruction: Branch to the instruction identified by the BranchToInstruction field in the executing instruction. As shown in **Figure 2**, an instruction is loaded into an instruction execution register **34**. The first instruction loaded is instruction zero. The ***NextConditions*** field of the current instruction is applied to a ***NextConditions*** compare block **36** which compares the NextConditions trigger signals. When the trigger signals satisfy the predetermined values in the current instruction, a NextConditions_True flag is set to logic 1. This causes the instruction address to be incremented and the next instruction in sequence to be loaded into instruction execution register **34**. Otherwise, the NextConditions_True flag is set to logic 0 and control passes to a Repeat Loop Conditions compare block **42**. Block **42** compares ~~compares~~ the content of the RepeatLoopControl field of the current instruction against the RepeatLoopControl decodes shown in **TABLE II**. If it matches one of the three RepeatLoopControl decodes, 01, 10, and 11, a RepeatLoop_Conditions_True signal is set to logic 1 and the instruction at the appropriate RepeatLoop_BranchTo-Instruction address is selected and the corresponding instruction is loaded into instruction execution register **34**. Otherwise, signal RepeatLoop_Conditions_True is set to logic 0 and the address specified in the BranchToInstruction field of the current instruction is loaded into the instruction address register and the corresponding instruction is loaded into instruction execution register **34**.

Replace paragraphs **[0048]** with the following paragraph:

[0048] Repeat loop module **26** provides optimal coding for redundant or symmetric sub-test sequences. Without the repeat loop module, a much larger number of instructions would be required to perform a memory test. The repeat loop module is used to repeat execution of a group of one or more sequential instructions. A group of sequential instructions includes the instructions between and including an instruction specified by a repeat loop module BranchToInstruction register and the

instruction which initiates the repeat operation. This group of sequential instructions is re-executed a plurality of times, with each instruction being modified in accordance with a set of modification commands for each repeat sequence or cycle. The repeat loop module may include one or more repeat loop circuits described later. The specific embodiment ~~illustrates~~ illustrated herein ~~include~~ includes two repeat loop circuits.